TRANSMITTAL OF APPEAL BRIEF (Large Entity)	Docket No. DE919990076

In Re Application Of: **Teich et al.**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
09/746,478	12/22/2000	Nadia Koshnoodi	46369	2137	8948

Invention: **METHOD AND SYSTEM FOR SECURELY MANAGING EEPROM DATA FILES**

COMMISSIONER FOR PATENTS:

Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed on:

March 2, 2006

The fee for filing this Appeal Brief is: **$500.00**

☐ A check in the amount of the fee is enclosed.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **09-0463 (IBM)**. I have enclosed a duplicate copy of this sheet.

☐ Payment by credit card. Form PTO-2038 is attached.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Kevin P. Radigan

Signature

Dated: **May 01, 2006**

Kevin P. Radigan, Esq.
Registration No.: 31,789

HESLIN ROTHENBERG FARLEY & MESITI, P.C.
5 Columbia Circle
Albany, New York 12203
Tel: (518) 452-5600
Fax: (518) 452-5579

cc:

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on **May 01, 2006**.

(Date)

Kevin P. Radigan

Signature of Person Mailing Correspondence

Kevin P. Radigan

Typed or Printed Name of Person Mailing Correspondence

AF
JFW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicants: Teich et al. Confirmation No.: 8948

Serial No.: 09/746,489 Group Art Unit: 2137

Filed: 12/22/2000 Examiner: Nadia Koshnoodi

Title: METHOD AND SYSTEM FOR SECURELY MANAGING EEPROM DATA FILES

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Mail Stop Appeal Briefs – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on May 01, 2006.

Kevin P. Radigan
Kevin P. Radigan
Attorney for Appellants
Registration No. 31,789

Date of Signature: May 01, 2006

Mail Stop Appeal Brief – Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Brief of Appellants

Dear Sir:

This is an appeal from a final rejection mailed December 2, 2005, rejecting claims 1-20 of the above-identified application, and subjecting claims 26-28 to a restriction requirement. This Brief is accompanied by a transmittal letter authorizing the charging of Appellants' deposit account for payment of the requisite fee set forth in 37 C.F.R. §1.17(c).

Appellants' Brief is believed to be in compliance with the requirements set forth in 37 C.F.R. §41.37(c). However, if Appellants' Brief does not comply with the requirements set forth in 37 C.F.R. §41.37(c), Appellants request notification of the reasons for non-compliance and the opportunity to file an amended Brief pursuant to 37 C.F.R. §41.37(d).

Real Party in Interest

This application is assigned to **International Business Machines Corporation** by virtue of an assignment executed by the co-inventors on December 6, 2000, and recorded with the United States Patent and Trademark Office at reel **011427**, frame **0535**, on December 22, 2000. Therefore, the real party in interest is **International Business Machines Corporation**.

Related Appeals and Interferences

To the knowledge of the Appellants, Appellants' undersigned legal representative, and the assignee, there are no other appeals or interferences, which will directly affect or be directly affected by or have a bearing on the Board's decision in the instant appeal.

Status of Claims

This patent application was filed on December 22, 2000, with the United States Patent and Trademark Office. As filed, the application included 25 claims, of which three (3) were independent (i.e., claims 1, 8 & 20).

In an initial Office Action dated November 8, 2004, claims 1-8 & 20-24 were rejected under 35 U.S.C. §102(b) as being anticipated by Schuyler (U.S. Patent No. 5,832,526; hereinafter Schuyler), while claims 9-11 & 25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schuyler in view of Steiner et al. (U.S. Patent No. 6,023,710; hereinafter Steiner), and claims 12-19 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schuyler in view of Steiner, and further in view of Kuo et al. (U.S. Patent No. 6,003,134; hereinafter Kuo). In Appellant's Response to Office Action dated March 8, 2005, claims 1, 6-9 & 20 were amended.

In a second Office Action mailed June 14, 2005, claims 1, 8 & 20-25 were objected to, claims 21-25 were rejected under 35 U.S.C. §101, and claims 21-22 were rejected under 35 U.S.C. §112, second paragraph. Additionally, claims 1-8 & 20-23 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schuyler and further in view of Klausmeier et al. (U.S. Patent No. 6,487,202; hereinafter Klausmeier) and Ofek et al. (U.S. Patent No. 5,751,993; hereinafter Ofek), while claims 9-11 & 25 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schuyler, Klausemeier and Ofek, and further in view of Steiner, and claims 12-19 & 24 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schuyler, Klausemeier, Ofek and Steiner, and further in view of Kuo. In Appellants' Response mailed September 8, 2005, claims 1-20 were amended, claims 21-25 were canceled, and new claims 26-28 were added.

In the final Office Action mailed December 2, 2005, claims 26-28 were withdrawn from consideration as being directed to a non-elected invention, and claims 1-8 & 20 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schuyler in view of Klausmeier in view of Ofek, while claims 9-11 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schuyler, Klausmeier, Ofek and further in view of Steiner, and claims 12-19 were rejected under 35 U.S.C. §103(a) as being unpatentable over Schuyler, Klausmeier, Ofek, Steiner and further in view of Kuo.

A Notice of Appeal to the Board of Patent Appeals and Interferences was mailed on March 2, 2006. The Notice of Appeal was received at the United States Patent and Trademark Office on March 6, 2006. The status of the claims is therefore as follows:

Claims allowed – none;

Claims objected to – none;

Claims rejected – 1-20;

Claims canceled – 21-25; and

Claims withdrawn – 26-28.

Appellants are appealing the rejection of claims 1-20 as stated in the final Office Action.

Status of Amendments

Appellants proffered no amendments responsive to the final Office Action of December 2, 2005. The claims as set out in the Claims Appendix include all prior entered claim amendments.

Summary of Claimed Subject Matter

In one aspect of the invention, Appellants claim a computer-implemented method (e.g., claim 1) (see, e.g., page 29, lines 19-31) which includes: securely managing an arbitrary number of data files in non-volatile storage (see, e.g., page 1, lines 9-13) in order to restore data after abortion of a write operation (see, e.g., page 4, line 22 – page 5, line 3), the data being stored in a record oriented data structure (see, e.g., FIGS. 1 & 2; page 4, lines 6-17; page 5, line 22 – page 6, line 7; page 9, lines 17-22) with each of the records containing, in addition to data contents, a first reference (Ptr. 3) indicating the current data-containing record of a previous file, and a second reference (Ptr. 1) indicating the current data-containing record of a subsequent file (see FIGS. 3A & 3B), wherein the write operation includes an update stage (see, e.g., page 6, line 8 – page 7, line 2) and an atomic write stage (see, e.g., page 7, line 3 – page 8, line 21), the update stage including multiple update operations performed for a plurality of records employing the second references of at least some records of the plurality of records (see, e.g., page 10, line 3 – page 13, line 16), and wherein the updates to the plurality of records are accepted in one atomic write stage after completion of the multiple update operations (see, e.g., page 13, lines 9-16), the one atomic write stage employing the first references of the at least some records of the plurality of records, and wherein at all times during the write operation, all the data files affected by the write operation contain either all the data stored prior to the write operation, or all the data as modified subsequent to the write operation (see, e.g., page 7, lines 3-14; page 10, line 3 – page 13, line 16).

In a further aspect of the invention, Appellants claim a computer-implemented method (e.g., claim 8) (see, e.g., page 29, lines 19-31) which includes: securely managing EEPROM data files in order to restore data after abortion of a write operation (see, e.g., page 4, line 22 – page 5, line 3), the data being stored in the files in a record-oriented data structure (see, e.g., FIGS. 1 & 2; page 4, lines 6-17; page 5, line 22 – page 6, line 7; page 9, lines 17-22), such that the data in

all files affected by the write operation is consistent with respect to completion of the write operation (see, e.g., page 7, lines 3-14), and wherein information concerning the status of the location of the consistent data is stored in the record oriented data structure together with the data (see, e.g., page 4, line 6 – page 5, line 3), wherein each record of the record oriented data structure of the files includes, in addition to data content, a first reference (Ptr. 3) indicating the current data-containing record of a previous file, and a second reference (Ptr. 1) indicating the current data-containing record of a subsequent file (see, e.g., FIGS. 3A & 3B), wherein the write operation includes an update stage (see, e.g., page 6, line 8 – page 7, line 2) and an atomic write stage (see, e.g., page 7, line 3 – page 8, line 21), the update stage including multiple update operations performed for a plurality of records employing the second references of at least some records of the plurality of records (see, e.g., page 10, line 3 – page 13, line 16), and wherein the atomic write stage is performed upon completion of the multiple update operations and employs the first references of the at least some records of the plurality of records (see, e.g., page 10, line 3 – page 13, line 16).

In a further aspect of the present invention, Appellants claim a system (e.g., claim 20) for securely managing EEPROM data files so that data can be restored after abortion of the write operation to the files (see, e.g., page 4, line 22 – page 5, line 3). The system includes an EEPROM and means for writing data to the EEPROM. The EEPROM includes a number of data files, each data file comprising a plurality of records in a record oriented data structure (see, e.g., FIGS. 1 & 2; page 4, lines 6-17; page 5, line 22 – page 6, line 7; page 9, lines 17-22), wherein each record of the record oriented data structure of the data files comprises, in addition to data content, a first reference (Ptr. 3) indicating the current data-containing record of a previous file, and a second reference (Ptr. 1) indicating the current data-containing record of a subsequent file (see FIGS. 3A & 3B), wherein the write operation includes an update stage (see, e.g., page 6, line 8 – page 7, line 2) and an atomic write stage (see, e.g., page 7, line 3 – page 8, line 21), the update stage comprising multiple update operations performed for a plurality of records employing the second references of the plurality of records, and wherein the atomic write stage is performed upon completion of the multiple update operations and employs the first references of the at least some records of the plurality of records (see, e.g., page 10, line 3 – page 13, line 16).

As a yet further aspect, dependent claim 9 recites that two or more data files are affected by the write operation (see, e.g., page 8, line 22 – page 9, line 16), and that new or modified data is written into the files in a cyclic manner, wherein each file comprises an indication of the number of records contained in the file and a plurality of records (FIG. 1), and wherein each record includes an indication of the status of the data in the record, and a synchronization number synchronizing with records of other files (see, e.g., FIG. 2; page 4, lines 6-17; and page 10, line 3 – page 13, line 16).

In another aspect, dependent claim 10 further recites: determining a current active record of a first of the files, and a working record of the first file (see, e.g., page 6, lines 10 & 11); setting the synchronization number of the working record of the file to the synchronization number of the current active record (see, e.g., page 6, lines 12-14); copying the data stored in the current active record into the working record and adding to or modifying the data according to the write operation, in the working record (see, e.g., page 6, lines 16 & 17); changing the status of the working record to the file to "active" (see, e.g., page 6, lines 18 & 19); repeating the steps for each further file (see, e.g., page 6, lines 21 & 22); and changing the record status of the original current active record of the first file to "inactive" as an indication that the write operation is complete (see, e.g., page 7, lines 1 & 2).

In yet another aspect, dependent claim 11 recites that the step of determining the current active record and the working record of the files includes searching for the first record in the file whose status byte indicates "active" status, and setting this record as the current active record, and setting the subsequent record as the working record (see, e.g., page 7, lines 15-23).

In yet another aspect, dependent claim 12 further recites: adding to or modifying the data of a record in the first file by identifying the current active record in the file and a working record and copying the data to be added to or modified from the current active record to the working record; modifying the data in the working record in accordance with the write operation, wherein the status byte of the current active record indicates that that record is "fully active" (see, e.g., page 10, lines 5-16) and the status of the working record indicates that that record is "inactive" (see, e.g., page 10, lines 17-19); setting synchronization indicator pointers to indicate that the file is the first file and to indicate that no further files have been modified (see, e.g., page

10, line 20 – page 11, line 2); identifying a current active record and a working record of a second file and copying the data from the current active record to the working record; modifying the data in the working record according to the write operation, wherein the status byte of the active current record indicates that the data in this record is "fully active" (see, e.g., page 11, lines 5-14) and the status byte of the working record indicates that this record is "inactive" (see, e.g., page 11, lines 15-17); setting synchronization indicator pointers to indicate the link between this file and the first file, and changing the synchronization indicator pointer of the first file to indicate its link with the second file (see, e.g., page 11, lines 17-23); and repeating the above steps for the second file for any subsequent files, wherein for the last file affected by the write operation, after setting the synchronization indicator pointers, determining that this is the last file, setting an indication pointer to indicate that no subsequent files are affected by the writing operation; and setting the status byte of each of the working records of the affected files to a "fully active" state, whereupon the write operation is complete and the modified data is the active data in all files (see, e.g., page 11, line 24 – page 13, line 16).

Grounds of Rejection to Be Reviewed On Appeal

1. Whether claims 1-8 & 20 were rendered obvious under 35 U.S.C. §103(a) to one of ordinary skill in the art by Schuyler, Klausmeier and Ofek.

2. Whether claims 9-11 were rendered obvious under 35 U.S.C. §103(a) to one of ordinary skill in the art by Schuyler, Klausmeier, Ofek and Steiner.

3. Whether claims 12-19 were rendered obvious under 35 U.S.C. §103(a) to one of ordinary skill in the art by Schuyler, Klausmeier, Ofek, Steiner and Kuo.

Argument

I. Rejection under 35 U.S.C. §103(a) over U.S. Patent No. 5,832,526 (to Schuyler) in view of U.S. Patent No. 6,487,202 (to Klausmeier) and U.S. Patent No. 5,751,993 (to Ofek)

Reversal of the rejection to claims 1-8 & 20 as obvious over Schuyler in view of Klausmeier and further in view of Ofek, is respectfully requested.

Advantageously, Appellants' record-oriented data structure with the doubly-linked list ensures that at all times during the write operation all the data files affected by the write operation contain either all the records stored prior to the write operation or all the data as modified subsequent to the write operation. In Appellants' approach, synchronicity of updates across records is guaranteed and a single atomic write stage is employed to finalize multiple updates to the plurality of records. Appellants' doubly-linked list structure provides a mechanism which allows data to be maintained as existing prior to an update operation until the single atomic write operation has been performed. This ensures the ability to recover from an interruption during the write operation.

In contrast, Schuyler discloses a method and apparatus for using slack area of file storage structures for file reconstruction. The file-reconstruction data includes a unique end-of-file tag code positioned within a vital core section of limited length. The vital core section is located close to or against the end of the slack area. Less-essential file-reconstruction data is grown frontwards from the vital core section to the beginning of the slack area as room permits. During file recovery, the random access storage device is scanned for storage sub-areas containing the predefined, unique end-of-file tag code appropriately positioned within the vital core portion at the end of the sub-area. Each storage sub-area having such a properly-positioned end-of-file tag code is designated as a possible end-of-file sub-area. A file recovery program uses this and other available information to reconstruct the directory structure to whatever extent is made possible. (See Abstract of Schuyler.)

Although Schuyler describes various file recovery techniques, a careful reading of Schuyler fails to uncover any teaching or suggestion of a write operation approach such as recited by Appellants in the independent claims presented herein. Specifically, there is no teaching or suggestion in Schuyler of a record-oriented data structure with each of the records which contains, in addition to the data contents, a first reference indicating the current data-containing record of a previous file, and a second reference indicating the current data-containing record of a subsequent file. Appellants' recited record-oriented data structure is essentially a doubly-linked list, including both a backward reference to the current data-containing record of a previous file, and a forward reference to the current data-containing record of a subsequent file.

There is simply no backward linking in the data structure described by Schuyler. Further, Appellants respectfully submit that a careful reading of Schuyler fails to uncover any express reference to even forward linking as recited by Appellants. In Appellants' recited invention, the record-oriented data structure includes a second reference which indicates the current data-containing record of a subsequent file.

Still further, Schuyler does not teach or suggest provision of a write operation as recited by Appellants in the independent claims. In Appellants' write operation, a first update stage is employed, followed by a second atomic write stage. In the update stage, multiple write operations are performed for a plurality of records employing the second references of at least some records of the plurality of records. After completion of the multiple update operations, Appellants' write operation accepts the updates to the plurality of records in one atomic write stage. The atomic write stage employs the first references of the at least some records of the plurality of records. Thus, in accordance with Appellants' write operation, there is a forward linked updating of the records, and a backward linked acceptance of the updates to the records. No similar functionality is believed taught or suggested by Schuyler or the other art of record.

The final Office Action recognizes certain of the above-noted deficiencies of Schuyler when applied against Appellants' independent claims. Specifically, at page 6, lines 9-15, the Office Action recognizes that Schuyler does not disclose records containing a second reference indicating the current data-containing record of a subsequent file, wherein the write operation comprises an update stage and an atomic write stage, the update stage comprising multiple update operations performed for a plurality of records employing the second references of at least some records of the plurality of records, and wherein the updates to the plurality of records are accepted by one atomic write stage after completion of the multiple update operations. Klausmeier and Ofek are cited as allegedly disclosing these aspects of Appellants' claimed process. This conclusion is respectfully traversed. The conclusion is believed based, in part, on a mischaracterization of Klausmeier's teachings at column 9, lines 17-41 (cited in the Office Action).

Klausmeier discloses a method of executing a sequence of multiple dependent operations, each operation including a memory read and a memory write involving overlapping memory accesses of the operations by grouping together memory reads and memory writes of multiple operations and preserving a desired sequence of the operations using a circuit external to a memory through which the memory accesses are performed. The operations may be updates to one or more linked lists. In one embodiment, the step of overlapping memory accesses may be performed by grouping together memory accesses according to ATM cell arrivals or departures. In this embodiment, the operations are associated with ATM cell arrivals or departures and may be GETs or PUTs, each GET and PUT operation may be characterized by a number of atomic memory operations to update one or more link lists. To perform the operations, a circuit is provided having an address processor, a data processor coupled to the address processor and to the external memory, and a prefetch buffer coupled to the external memory, the address processor and the data processor. The address processor generates memory addresses for the operations according to the steps of overlapping memory accesses. The atomic memory operations are grouped so that all of the memory read operations associated with the dependent operations are performed before all of the memory write operations associated with the dependent operations are performed. (See Abstract of Klausmeier.)

Klausmeier depicts in Fig. 6 a queue controller, which is described, in part, at column 8, line 12 – column 10, line 3, with column 9, line 17-29 & lines 27-41 being specifically cited in the Office Action. A careful reading of this Klausmeier material fails to uncover any relevancy to Appellants' claimed process.

Specifically, column 9, lines 17-29 of Klausmeir state:

> In order for head array 704, tail array 706 and queue array 708 to reflect the current state of cell memory 622, Queue Controller 600 should be informed every time a cell is stored in cell memory 622 (cell arrival event) and every time a cell is to be transmitted from cell memory 622 (cell departure event). In addition, the above-discussed arrays must be updated for each cell move event (a cell move occurs when a cell is moved from a VC queue to a QBin queue – in fact, the cell is not physically moved with cell memory 622, however, various linked list pointers for each queue involved in the move event must be updated). Such updating requires a number of atomic operations for each cell event.

Appellants respectfully submit that there is no teaching or suggestion in the above-noted lines of Klausmeier for a record-oriented data structure such as recited by Appellants in the independent claims. In Appellants' invention, each of the records contains, in addition to data contents, a first reference indicating the current data-containing record of a previous file, and a second reference indicating the current data-containing record of a subsequent file. This information is then employed in a write operation that includes two stages, an update stage and an atomic write stage. During the update stage, multiple update operations are performed for a plurality of records employing the second references of at least some records of the plurality of records. These updates to the plurality of records are then accepted in one atomic write stage after completion of the multiple update operations. No similar processing functionality is taught or suggested by Klausmeier. The Office Action provides no discussion as to how the above-noted teaching of Klausmeier can be extrapolated and made relevant to Appellants' claimed process.

The Office Action also alleges that Klausmeier teaches Appellants' recited write operation comprising two stages, with updates accepted in a single atomic write stage. This assertion is believed clearly contrary to the teachings of Klausmeier, i.e., to any extent Klausmeier is relevant to Appellants' claimed process. For example, Klausmeier teaches at column 9, lines 28-30, that "such updating requires a number of atomic memory operations for each cell event." Thus, not only is the underlying disclosure not relevant to Appellants' claimed process, but this extrapolated teaching of Klausmeier is believed contrary to the express description provided therein.

The Office Action further recognizes that Schuyler does not teach that each record contains a first reference indicating the current data-containing record of a previous file, with the one atomic stage employing the first reference of the at least some records of the plurality of records. For this aspect of Appellants' invention, Ofek is cited. Specifically, Ofek notes at column 2, lines 9-19, the existence of a queue implemented as a doubly-linked list which is employed by a cache manager to remove or replace the "least-recently-used" data element in the cache memory. This is clearly a different process than that recited by Appellants. In Appellants' invention, the doubly-linked lists comprising the first references and second references of each record are employed in a write operation that includes two stages. An update stage and a single

atomic write stage. The update stage employs the second references of the at least some records of the plurality of records, while the single atomic write stage employs the first references of the at least some records of the plurality of records. No similar processing is taught or suggested by Schuyler, Klausmeier and/or Ofek, singularly or in combination. There is simply no suggestion in these three documents of a process such as recited by Appellants in the independent claims. Thus, in addition to the Office Action misinterpreting the teachings of Klausmeier, Appellants respectfully submit that the documents combined fail to disclose Appellants' claimed process.

Further, Appellants respectfully traverse the combinability of Klausmeier and Ofek with Schuyler as alleged in the Office Action. For example, with respect to Klausmeier, the Office Action provides no rationale for combining Klausmeier with Schuyler, except to indicate that the rationale is suggested by Klausmeier at column 9, lines 17-29, in one instance, and at column 9, lines 27-41, in the other instance. These citations are believed deficient as a basis for alleging the combination of the two documents. Further, as noted by Appellants above, the cited lines are believed mischaracterized in the Office Action, and to any extent applicable, actually teach away from Appellants' claimed process. A careful reading of the cited lines fails to uncover any suggestion that would lead one skilled in the art to extrapolate a teaching such as noted in the Office Action and apply that teaching to Schuyler. Further, the Office Action provides no explanation as to why the cited lines of Klausmeier suggest the combination. Absent such an explanation, Appellants respectfully submit that the Office Action is deficient and requests reconsideration and withdrawal of the obviousness rejection to the independent claims presented based thereon.

Further, the combination of Ofek with Klausmeier and Schuyler is also believed deficient for the above-noted reasons. The Office Action fails to provide any explanation as to why one of ordinary skill in the art would have combined Ofek with Schuyler and Klausmeier. The only motivation indicated by the Examiner is a citation to column 2, lines 9-19 of Ofek, which describe a cache manager and a queue, with the queue being implemented as a doubly-linked list. However, Appellants are not claiming simply a doubly-linked list, but rather, claim a specific write operation process which includes two stages, an update stage and an atomic write stage, with one of the references being employed in the update stage, and the other of the references being employed in the atomic write stage. No similar process functionality is believed taught or

suggested by Ofek at column 2, lines 9-19, notwithstanding the teaching of a queue comprising a doubly-linked list. The Office Action provides no explanation as to how one of ordinary skill in the art would go about combining the teachings of Ofek, and Klausmeier into Schuyler to arrive at Appellants' claimed process. Absent this showing, it is respectfully submitted that a *prima facie* case of obviousness is not stated in the Office Action against the independent claims presented.

Still further, Appellants respectfully submit that upon a review of Schuyler, Klausmeier and Ofek, there is no suggestion, or incentive for combining and further modifying the processing thereof as would be necessary to achieve Appellants' invention. To the extent addressed in the Office Action, the characterizations and teachings of, particularly, Klausmeier and Ofek, set forth no technical basis outside that contained in Appellants' own specification for the extrapolated teachings cited therein and the combinability thereof with Schuyler to arrive at Appellants' claimed invention. In this aspect, Appellants respectfully submit that the characterizations of the teachings of Klausmeier and Ofek merely assert language of Appellants' claimed invention in hindsight. Thus, the rejection violates the well-known principle that Appellants' own disclosure cannot be used as a reference against them.

For at least the above-noted reasons, Appellants respectfully request reversal of the obviousness rejection to claims 1-8 & 20 based on Schuyler in view of Klausmeier and Ofek.

II. Rejection under 35 U.S.C. §103(a) over U.S. Patent No. 5,832,526 (to Schuyler), in view of U.S. Patent No. 6,487,202 (to Klausmeier), U.S. Patent No. 5,751,993 (to Ofek), and U.S. Patent No. 6,023,710 (to Steiner)

Reversal of the rejection to claims 9-11 as obvious over Schuyler, Klausmeier, Ofek and Steiner is respectfully requested.

Dependent claims 9-11 further characterize independent claim 8 discussed above in connection with Argument I. Thus, these claims are believed allowable initially for the reasons stated above with respect to the independent claim.

Further, the claims at issue provide additional detail on Applicants' recited method which are believed clearly not taught or suggested by the applied art. For example, claim 9 recites that each record includes an indication of the status of the data in the record, and a synchronization

number synchronizing with records of other files. A careful reading of the cited lines of Steiner fails to uncover any teaching or suggestion of including an indication within each record of the status of the data in the record. As such, it is respectfully submitted that the Office Action fails to state a *prima facie* case of obviousness against the subject matter of claims 9-11.

By way of further example, claim 10 provides specific characterization of the determining step recited in Appellants' claim. These characterizations include: determining a current active record of a first of the files, and a working record of the first file; setting the synchronization number of the working record of the file to the synchronization number of the current active record; copying the data stored in the current active record into the working record and adding to or modifying the data according to the write operation, in the working record; changing the status of the working record of the file to "active", repeating the steps for each further file, and changing the record status of the original current active record of the first file to "inactive" as an indication that the write operation is complete.

For an alleged teaching or suggestion of these steps, the Office Action relies on Steiner, and principally column 3, lines 19-41 & lines 51-61 thereof. However, Appellants respectfully submit that a careful reading of these lines fails to disclose any relevancy to Appellants' recited process. Steiner is describing a process of coalescing various pieces from different storage sessions into a single file. In contrast, Appellants are reciting a process for performing a write operation. The particular steps outlined above in Appellants' recited dependent claim 10 are believed clearly distinct from the process of coalescing different portions of a file described by Steiner at column 3. For example, there is no discussion of a working record or a current active record in Steiner, there is no discussion of setting a synchronization number of a working record of the file to a synchronization number of the current active record, there is no discussion of changing the status of the working record of the file to "active", or of changing the record status of the original current active record of the first file to "inactive" as an indication that the write operation is complete.

For at least the above-noted reasons, Appellants respectfully request reversal of the obviousness rejection to claims 9-11 based on Schuyler in view of Klausmeier, Ofek and Steiner.

III. Rejection under 35 U.S.C. §103(a) over U.S. Patent No. . 5,832,526 (to Schuyler), in view of U.S. Patent No. 6,487,202 (to Klausmeier), U.S. Patent No. 5,751,993 (to Ofek), U.S. Patent No. 6,023,710 (to Steiner), and U.S. Patent No. 6,003,134 (to Kuo)

Reversal of the rejection to claims 12-19 as obvious over Schuyler, Klausmeier, Ofek, Steiner and Kuo is respectfully requested.

Claim 12 sets forth a detailed process which builds on claims 8, 9, 10 & 11, and which includes characterizations of how to add or modify data of a record in a first file employing a current active record, a working record, a status byte for the current active record, a status byte for the working record, and synchronization indicator pointers indicating links between files. After a last file is determined, the process includes setting an indication pointer to indicate that no subsequent files are affected by the writing operation, and setting the status byte of each of the working records in the affected files to a "fully active" state, thereby completing the write operation, meaning that the modified data is the active data in all the files. Appellants respectfully submit that a careful reading of Schuyler, Klausmeier, Ofek, Steiner and Kuo fails to provide any teaching or suggestion of the particular process recited by Appellants in these claims.

In this regard, Appellants note that the Office Action fails to specifically address each aspect of Appellants' recited functionality when comparing the claims at issue to the applied art. As such, it is respectfully submitted that the Office Action fails to state a *prima facie* case of obviousness against claim 12. At page 18 of the Office Action, the Kuo patent is added for a teaching of storing signature-type information that may be stored with a file and identifying which copies of a file are current. (Column 12, lines 31-46.) However, Appellants' invention is not simply directed to identifying whether a file, a current active record, or a working record are current, but rather to a specific process for achieving multiple updates in combination with a single write operation as set forth.

For at least the above-noted reason, Appellants respectfully request reversal of the obviousness rejection to claims 12-19 based on Schuyler in view of Klausmeier, Ofek, Steiner and Kuo.

Conclusion

Appellants respectfully request reversal of the 35 U.S.C. §103(a) rejections to claims 1-20 based on Schulyer, Klausmeier, Ofek, Steiner and Kou. Appellants respectfully submit that Schuyler, Klausmeier and Ofek would not have rendered their claimed invention obvious to one of ordinary skill in the art. For example, these patents do not, individually or in combination, teach or suggest at least Appellants' recited independent claims, which include: securely managing an arbitrary number of data files in non-volatile storage in order to restore data after abortion of a write operation. The write operation includes an update stage and an atomic write stage. The update stage includes multiple update operations performed for a plurality of records employing the second references of at least some records of the plurality of records, wherein the updates to the plurality of records are accepted in one atomic write stage after completion of the multiple update operations. The one atomic write stage employs the first references of the at least some references of the plurality of records. At all times during the write operation, all the data files affected by the write operation contain either all the data stored prior to the write operation, or all the data as modified subsequent to the write operation.

Accordingly, reversal of all rejections is respectfully requested.

Kevin P. Radigan

Kevin P. Radigan
Reg. No. 31,789
Attorney for Appellants

Dated: May 01, 2006

HESLIN ROTHENBERG FARLEY & MESITI P.C.
5 Columbia Circle
Albany, New York 12203-5160
Telephone: (518) 452-5600
Facsimile: (518) 452-5579

Claims Appendix

1. A computer-implemented method comprising:

securely managing an arbitrary number of data files in non-volatile storage in order to restore data after abortion of a write operation, the data being stored in a record oriented data structure with each of the records containing, in addition to data contents, a first reference indicating the current data-containing record of a previous file, and a second reference indicating the current data-containing record of a subsequent file, wherein the write operation comprises an update stage and an atomic write stage, the update stage comprising multiple update operations performed for a plurality of records employing the second references of at least some records of the plurality of records, and wherein the updates to the plurality of records are accepted in one atomic write stage after completion of the multiple update operations, the one atomic write stage employing the first references of the at least some records of the plurality of records, and wherein at all times during the write operation, all of the data files affected by the write operation contain either all of the data stored prior to the write operation, or all of the data as modified subsequent to the write operation.

2. The computer-implemented method as claimed in claim 1, wherein each file affected by a write operation comprises a plurality of records, one of said records in each file containing the data prior to said write operation and another of said records being modified according to said write operation, each of said records also comprising a status byte indicative of the status of the data contained therein.

3. The computer-implemented method as claimed in claim 2, wherein said data prior to the write operation, in each file, is retained as the active data in the case of a power failure, until all files have been successfully updated according to said write operation.

4. The computer-implemented method as claimed in claim 1, wherein each record contains a synchronization byte, indicating a relationship with records of other files.

5. The computer-implemented method as claimed in claim 3, wherein each record contains a synchronization byte, indicating a relationship with records of other files.

6. The computer-implemented method as claimed in claim 5, wherein the first reference of each record comprises a first pointer (PTR 1) indicating the current data-containing record of a previous file and the second reference of each record comprises a further pointer (PTR 3) indicating the current data-containing record of a subsequent file.

7. The computer-implemented method as claimed in claim 6, wherein a third reference of each record comprises a second pointer (PTR 2) indicating the current data-containing record of that file.

8. A computer-implemented method comprising:

securely managing EEPROM data files in order to restore data after abortion of a write operation, the data being stored in the files in a record-oriented data structure, such that the data in all files affected by the write operation is consistent with respect to completion of the write operation, and wherein information concerning the status and location of the consistent data is stored in the record oriented data structure together with the data, wherein each record of the record oriented data structure of the files comprises, in addition to data contents, a first reference indicating the current data-containing record of a previous file, and a second reference indicating the current data-containing record of a subsequent file, wherein the write operation comprises an update stage and an atomic write stage, the update stage comprising multiple update operations performed for a plurality of records employing the second references of at least some records of the plurality of records, and wherein the atomic write stage is performed upon completion of the multiple update operations and employs the first references of the at least some records of the plurality of records.

9. The computer-implemented method according to claim 8, wherein two or more data files are affected by said write operation, and wherein new or modified data is written into said files in a cyclic manner, wherein each file comprises an indication of the number of records contained in said file and a plurality of records, and wherein each record comprises an indication of the status of the data in said record, and a synchronisation number synchronising with records of other files.

10. The computer-implemented method according to claim 9, further comprising determining a current active record of a first of said files, and a working record of said first file; setting the synchronization number of the working record of said file to the synchronization number of the current active record; copying the data stored in said current active record into said working record and adding to or modifying said data according to said write operation, in said working record; changing the status of said working record of said file to 'active'; repeating said steps for each further file; and changing the record status of said original current active record of said first file to 'inactive' as an indication that said write operation is complete.

11. The computer-implemented method as claimed in claim 10, wherein said step of determining the current active record and the working record of said files comprises searching for the first record in said file whose status byte indicates 'active' status and setting this record as said current active record, and setting the subsequent record as said working record.

12. The computer-implemented method as claimed in claim 11, comprising:

adding to or modifying the data of a record in the first file by:

identifying the current active record of said file and a working record and copying the data to be added to or modified from the current active record to the working record;

modifying the data in said working record in accordance with the write operation; wherein the status byte of said current active record indicates that that record is 'fully active' and the status of said working record indicates that that record is 'inactive';

setting synchronization indicator pointers to indicate that said file is said first file and to indicate that no further files have been modified;

identifying a current active record and a working record of a second file and copying the data from the current active record to the working record; modifying the data in the working record according to said write operation, wherein the status byte of said active current record indicates that the data in this record is "fully active" and the status byte of the working record indicates that this record is 'inactive';

setting synchronization indicator pointers to indicate the link between this file and said first file, and changing said synchronization indicator pointer of said first file to indicate its link with said second file; and

repeating these steps for said second file for any subsequent files, wherein

for the last file affected by said write operation, after setting said synchronization indicator pointers, determining that this is the last file, setting an indication pointer to indicate that no subsequent files are affected by said writing operation; and

setting the status byte of each of said working records of said affected files to a 'fully active' state, whereupon the write operation is complete and the modified data is the active data in all files.

13. The computer-implemented method as claimed in claim 12, wherein, upon interruption of said write operation at any stage, either all current active records of all files affected by said operation are set as 'fully active' records, and the data contained in said files prior to the start of said write operation is the current active data, or all working records of all files are set to a 'fully active' status, in which case all files contain the modified data due to said write operation as said active data.

14. The computer-implemented method as claimed in claim 13, wherein interruption of said write operation during or immediately after the step of modifying the data in the working record of said first file results in the current active record of said first file remaining as the 'fully active' data record, at which time no further files have been modified and all of the 'active' datable files correspond to the data prior to the write operation.

15. The computer-implemented method as claimed in claim 13, wherein an interruption of said write operation during or subsequent to the setting of the synchronization indicator pointers in said first file results in the current active record of said first file remaining as the 'fully active' data record, at which time no further files have been modified and all of the 'active' datable files correspond to the data prior to the write operation.

16. The computer-implemented method as claimed in claim 13, wherein an interruption of said write operation during or immediately after the step of modifying the data in the second or subsequent files results in the current active record of said second or subsequent file remaining set as said 'fully active' record, and, since said synchronization indicator pointer of said first file still indicates that said current active record is still said 'fully active' record of said first file, the currently active data of both or all of said files remains as that prior to the start of the write operation.

17. The computer-implemented method as claimed in claim 13, wherein an interruption to said write process during or immediately after modifying the data in the working record of the last file affected by said write operation, results in all of the current active records of all of said files being retained as said fully active records, wherein the currently active data corresponds to the data prior to the write operation.

18. The computer-implemented method as claimed in claim 13, wherein an interruption of said write process during or immediately after modification of the data in the working record of the last file affected by said write operation, causes all working records of all of said files to become set to 'fully active' records, such that all files contain data modified as a result of said write operation as the currently active data.

19. The computer-implemented method as claimed in claim 12, wherein, when all of said write steps have been successfully completed, without an interruption, said synchronization indicator pointers are used to indicate the links between the modified records of the files affected, and all working records are set to status 'fully active' and said current active records are set to status 'inactive'.

20. A system for securely managing EEPROM data files so that the data can be restored after abortion of the write operation to said data files, the system comprising:

an EEPROM, and means for writing data to said EEPROM, said EEPROM comprising a number of data files, each data file comprising a plurality of records in a record oriented data structure, wherein each record of the record oriented data structure of the data files comprises, in addition to data content, a first reference indicating the current data-containing record of a previous file, and a second reference indicating the current data-containing record of a subsequent file, wherein the write operation comprises an update stage and an atomic write stage, the update stage comprising multiple update operations performed for a plurality of records employing the second references of the plurality of records, and wherein the atomic write stage is performed upon completion of the multiple update operations and employs the first references of the at least some records of the plurality of records.

21-25. (Canceled).

26. (Withdrawn) A computer-implemented method for ensuring the consistency of data stored in records in different files, the records being changed by a transaction, each change to a record of a file leading to the generation of new records in the files affected by the transaction, the method comprising:

a) designating one of the files affected by the transaction as a primary file, only the primary file having a status concerning the successful execution of the transaction, the status assuming the value "current" or "not current";

b) setting the status to "current" in each newly generated primary record in the primary file;

c) testing whether the primary file contains multiple primary records having the "current" status;

d) resetting the status of the primary record of the primary file preceding the newly generated primary record to "not current", and retaining the "current" status of the newly generated primary record, if the write operation for a record of the files affected by the transaction is determined not to have terminated prematurely and resetting the status of the newly generated primary record to "not current", and retaining the "current" status of the preceding primary record of the primary file, if the write operation for a record of the files affected by the transaction is determined to have terminated prematurely.

27. (Withdrawn) The computer-implemented method of claim 26, wherein each primary record of the primary file contains data for identifying records in files affected by the transaction.

28. (Withdrawn) The computer-implemented method of claim 27, wherein the records of the primary file and the records of the files affected by the transaction are linked by a pointer.

* * * * *

Evidence Appendix

None.

Related Proceedings Appendix

None.